(Home)

THE DARPA INTERNET GATEWAY

RFC 823

Robert Hinden
Alan Sheltzer

Bolt Beranek and Newman Inc.
10 Moulton St.
Cambridge, Massachusetts 02238

September 1982

Prepared for

Defense Advanced Research Projects Agency
Information Processing Techniques Office
1400 Wilson Boulevard
Arlington, Virginia 22209

This RFC is a status report on the Internet Gateway developed by BBN. It describes the Internet Gateway as of September 1982. This memo presents detailed descriptions of message formats and gateway procedures, however this is not an implementation specification, and such details are subject to change.

Table of Contents

1 INTRODUCTION..... 1
2 BACKGROUND..... 2
3 FORWARDING INTERNET DATAGRAMS..... 5
3.1 Input..... 5
3.2 IP Header Checks..... 6
3.3 Routing..... 7
3.4 Redirects..... 9
3.5 Fragmentation..... 9
3.6 Header Rebuild..... 10
3.7 Output..... 10
4 PROTOCOLS SUPPORTED BY THE GATEWAY..... 12
4.1 Cross-Net Debugging Protocol..... 12
4.2 Host Monitoring Protocol..... 12
4.3 ICMP..... 14
4.4 Gateway-to-Gateway Protocol..... 14
4.4.1 Determining Connectivity to Networks..... 14
4.4.2 Determining Connectivity to Neighbors..... 16
4.4.3 Exchanging Routing Information..... 17
4.4.4 Computing Routes..... 19
4.4.5 Non-Routing Gateways..... 22
4.4.6 Adding New Neighbors and Networks..... 23
4.5 Exterior Gateway Protocol..... 24
5 GATEWAY SOFTWARE..... 26
5.1 Software Structure..... 26
5.1.1 Device Drivers..... 27
5.1.2 Network Software..... 27
5.1.3 Shared Gateway Software..... 29
5.2 Gateway Processes..... 29
5.2.1 Network Processes..... 29
5.2.2 GGP Process..... 30
5.2.3 HMP Process..... 31
APPENDIX A. GGP Message Formats..... 32
APPENDIX B. Information Maintained by Gateways..... 39
APPENDIX C. GGP Events and Responses..... 41
REFERENCES..... 43

1 INTRODUCTION

This document explains the design of the Internet gateway used in the Defense Advanced Research Project Agency (DARPA) Internet program. The gateway design was originally documented in IEN-30, "Gateway Routing: An Implementation Specification" [2], and was later updated in IEN-109, "How to Build a Gateway" [3]. This document reflects changes made both in the internet protocols and in the gateway design since these documents were released. It supersedes both IEN-30 and IEN-109.

The Internet gateway described in this document is based on the work of many people; in particular, special credit is given to V. Strazisar, M. Brescia, E. Rosen, and J. Haverty.

The gateway's primary purpose is to route internet datagrams to their destination networks. These datagrams are generated and processed as described in RFC 791, "Internet Protocol - DARPA Internet Program Protocol Specification" [1]. This document describes how the gateway forwards datagrams, the routing algorithm and protocol used to route them, and the software structure of the current gateway. The current gateway implementation is written in macro-11 assembly language and runs in the DEC PDP-11 or LSI-11 16-bit processor.

2 BACKGROUND

The gateway system has undergone a series of changes since its inception, and it is continuing to evolve as research proceeds in the Internet community. This document describes the implementation as of mid-1982.

Early versions of gateway software were implemented using the BCPL language and the ELF operating system. This implementation evolved into one which used the MOS operating system for increased performance. In late 1981, we began an effort to produce a totally new gateway implementation. The primary motivation for this was the need for a system oriented towards the requirements of an operational communications facility, rather than the research testbed environment which was associated with the BCPL implementation. In addition, it was generally recognized that the complexity and buffering requirements of future gateway configurations were beyond the capabilities of the PDP-11/LSI-11 and BCPL architecture. The new gateway implementation therefore had a second goal of producing a highly space-efficient implementation in order to provide space for buffers and for the extra mechanisms, such as monitoring, which are needed for an operational environment.

This document describes the implementation of this new gateway which incorporates several mechanisms for operations activities, is coded in assembly language for maximum space-efficiency, but otherwise is fundamentally the same architecture as the older, research-oriented, implementations.

One of the results of recent research is the thesis that gateways should be viewed as elements of a gateway system, where the gateways act as a loosely-coupled packet-switching communications system. For reasons of maintainability and operability, it is easiest to build such a system in an homogeneous fashion where all gateways are under a single authority and control, as is the practice in other network implementations.

In order to create a system architecture that permitted multiple sets of gateways with each set under single control but acting together to implement a composite single Internet System, new protocols needed to be developed. These protocols, such as the "Exterior Gateway Protocol," will be introduced in the later releases of the gateway implementation.

We also anticipate further changes to the gateway architecture and implementation to introduce support for new

capabilities, such as large numbers of networks, access control, and other requirements which have been proposed by the Internet research community. This document represents a snapshot of the current implementation, rather than a specification.

3 FORWARDING INTERNET DATAGRAMS

This section describes how the gateway forwards datagrams between networks. A host computer that wants an IP datagram to reach a host on another network must send the datagram to a gateway to be forwarded. Before it is sent into the network, the host attaches to the datagram a local network header containing the address of the gateway.

3.1 Input

When a gateway receives a message, the gateway checks the message's local network header for possible errors and performs any actions required by the host-to-network protocol. This processing involves functions such as verifying the local network header checksum or generating a local network acknowledgment message. If the header indicates that the message contains an Internet datagram, the datagram is passed to the Internet header check routine. All other messages received that do not pass these tests are discarded.

3.2 IP Header Checks

The Internet header check routine performs a number of validity tests on the IP header. Datagrams that fail these tests are discarded causing an HMP trap to be sent to the Internet Network Operations Center (INOC) [7]. The following checks are currently performed:

- Proper IP Version Number
- Valid IP Header Length (>= 20 bytes)
- Valid IP Message Length
- Valid IP Header Checksum
- Non-Zero Time to Live field

After a datagram passes these checks, its Internet destination address is examined to determine if the datagram is addressed to the gateway. Each of the gateway's internet addresses (one for each network interface) is checked against the destination address in the datagram. If a match is not found, the datagram is passed to the forwarding routine.

If the datagram is addressed to the gateway itself, the IP options in the IP header are processed. Currently, the gateway supports the following IP options:

- o NOP
- o End of Option List
- o Loose Source and Record Route
- o Strict Source and Record Route

The datagram is next processed according to the protocol in the IP header. If the protocol is not supported by the gateway, it replies with an ICMP error message and discards the datagram. The gateway does not support IP reassembly, so fragmented datagrams which are addressed to the gateway are discarded.

3.3 Routing

The gateway must make a routing decision for all datagrams that are to be to forwarded. The routing algorithm provides two pieces of information for the gateway: 1) the network interface that should be used to send this datagram and 2) the destination address that should be put in the local network header of the datagram.

The gateway maintains a dynamic Routing Table which contains an entry for each reachable network. The entry consists of a network number and the address of the neighbor gateway on the shortest route to the network, or else an indication that the

gateway is directly connected to the network. A neighbor gateway is one which shares a common network with this gateway. The distance metric that is used to determine which neighbor is closest is defined as the "number of hops," where a gateway is considered to be zero hops from its directly connected networks, one hop from a network that is reachable via one other gateway, etc. The Gateway-to-Gateway Protocol (GGP) is used to update the Routing Table (see Section 4.4 describing the Gateway-to-Gateway Protocol).

The gateway tries to match the destination network address in the IP header of the datagram to be forwarded, with a network in its Routing Table. If no match is found, the gateway drops the datagram and sends an ICMP Destination Unreachable message to the IP source. If the gateway does find an entry for the network in its table, it will use the network address of the neighbor gateway entry as the local network destination address of the datagram. However, if the final destination network is one that the gateway is directly connected to, the destination address in the local network header is created from the destination address in the IP header of the datagram.

3.4 Redirects

If the routing procedure decides that an IP datagram is to be sent back out the same network interface that it was read in, then this gateway is not on the shortest path to the IP final destination. Nevertheless, the datagram will still be forwarded to the next address chosen by the routing procedure. If the datagram is not using the IP Source Route Option, and the IP source network of the datagram is the same as the network of the next gateway chosen by the routing procedure, an ICMP Redirect message will be sent to the IP source host indicating that another gateway should be used to send traffic to the final IP destination.

3.5 Fragmentation

The datagram is passed to the fragmentation routine after the routing decision has been made. If the next network through which the datagram must pass has a maximum message size that is smaller than the size of the datagram, the datagram must be fragmented. Fragmentation is performed according to the algorithm described in the Internet Protocol Specification [1]. Certain IP options must be copied into the IP header of all

fragments, and others appear only in the first fragment according to the IP specification. If a datagram must be fragmented, but the Don't fragment bit is set, the datagram is discarded and an ICMP error message is sent to the IP source of the datagram.

3.6 Header Rebuild

The datagram (or the fragments of the original datagram if fragmentation was needed) is next passed to a routine that rebuilds the Internet header. The Time to Live field is decremented by one and the IP checksum is recomputed.

The local network header is now built. Using the information obtained from its routing procedure, the gateway chooses the network interface it considers proper to send the datagram and to build the destination address in the local network header.

3.7 Output

The datagram is now enqueued on an output queue for delivery towards its destination. A limit is enforced on the size of the output queue for each network interface so that a slow network

does not unfairly use up all of the gateway's buffers. If a datagram cannot be enqueued due to the limit on the output queue length, it is dropped and an HMP trap is sent to the INOC. These traps, and others of a similar nature, are used by operational personnel to monitor the operations of the gateways.

4 PROTOCOLS SUPPORTED BY THE GATEWAY

A number of protocols are supported by the gateway to provide dynamic routing, monitoring, debugging, and error reporting. These protocols are described below.

4.1 Cross-Net Debugging Protocol

The Cross-Net Debugging Protocol (XNET) [8] is used to load the gateway and to examine and deposit data. The gateway supports the following XNET op-codes:

- o NOP
- o Debug
- o End Debug
- o Deposit
- o Examine
- o Create Process

4.2 Host Monitoring Protocol

The Host Monitoring Protocol (HMP) [6] is used to collect measurements and status information from the gateways. Exceptional conditions in the gateways are reported in HMP traps. The status of a gateway's interfaces, neighbors, and the networks which it can reach are reported in the HMP status message.

Two types of gateway statistics, the Host Traffic Matrix and the gateway throughput, are currently defined by the HMP. The Host Traffic Matrix records the number of datagrams that pass through the gateway with a given IP source, destination, and protocol number. The gateway throughput message collects a number of important counters that are kept by the gateway. The current gateway reports the following values:

- Datagrams dropped because destination net unreachable
- Datagrams dropped because destination host unreachable
- Per Interface:
  - Datagrams received with IP errors
  - Datagrams received for this gateway
  - Datagrams received to be forwarded
  - Datagrams looped
  - Bytes received
  - Datagrams sent, originating at this gateway
  - Datagrams sent to destination hosts
  - Datagrams dropped due to flow control limitations
  - Datagrams dropped due to full queue
  - Bytes sent
- Per Neighbor:
  - Routing updates sent to
  - Routing updates received from
  - Datagrams sent, originating here
  - Datagrams forwarded to
  - Datagrams dropped due to flow control limitations
  - Datagrams dropped due to full queue
  - Bytes sent

4.3 ICMP

The gateway will generate the following ICMP messages under appropriate circumstances as defined by the ICMP specification [4]:

- o Echo Reply
- o Destination Unreachable
- o Source Quench
- o Redirect
- o Time Exceeded
- o Parameter Problem
- o Information Reply

4.4 Gateway-to-Gateway Protocol

The gateway uses the Gateway-to-Gateway Protocol (GGP) to determine connectivity to networks and neighbor gateways; it is also used in the implementation of a dynamic, shortest-path routing algorithm. The current GGP message formats (for release 1003 of the gateway software) are presented in Appendix A.

4.4.1 Determining Connectivity to Networks

When a gateway starts running it assumes that all its neighbor gateways are "down," that it is disconnected from

networks to which it is attached, and that the distance reported in routing updates from each neighbor to each network is "infinity."

The gateway first determines the state of its connectivity to networks to which it is physically attached. The gateway's connection to a network is declared up if it can send and receive internet datagrams on its interface to that network. Note that the method that the gateway uses to determine its connectivity to a network is network-dependent. In some networks, the host-to-network protocol determines whether or not datagrams can be sent and received on the host interface. In these networks, the gateway simply checks-status information provided by the protocol in order to determine if it can communicate with the network. In other networks, where the host-to-network protocols are less sophisticated, it may be necessary for the gateway to send datagrams to itself to determine if it can communicate with the network. In these networks, the gateways periodically poll the network using GGP network interface status messages [Appendix A] to determine if the network interface is operational.

The gateway has two rules relevant to computing distances to networks: 1) if the gateway can send and receive traffic on its

network interface, its distance to the network is zero; 2) if it cannot send and receive traffic on the interface, its distance to the network is "infinity." Note that if a gateway's network interface is not working, it may still be able to send traffic to the network on an alternate route via one of its neighbor gateways.

4.4.2 Determining Connectivity to Neighbors

The gateway determines connectivity to neighbors using a "K out of N" algorithm. Every 15 seconds, the gateway sends GGP Echo messages [Appendix A] to each of its neighbors. The neighbors respond by sending GGP echo replies. If there is no reply to K out of N (current values are K=3 and N=4) echo messages sent to a neighbor, the neighbor is declared down. If a neighbor is down and J out of M (current values are J=2 and M=4) echo replies are received, the neighbor is declared to be up. The values of J,K,M,N and the time interval are operational parameters which can be adjusted as required.

4.4.3 Exchanging Routing Information

The gateway sends routing information in GGP Routing Update messages. The gateway receives and transmits routing information reliably using sequence-numbered messages and a retransmission and acknowledgment scheme as explained below. For each neighbor, the gateway remembers the Receive Sequence Number, R, that it received in the most recent routing update from that neighbor. This value is initialized with the sequence number in the first Routing Update received from a neighbor after that neighbor's status is set to "up." On receipt of a routing update from a neighbor, the gateway subtracts the Receive Sequence Number, R, from the sequence number in the routing update, S. If this value (S-R) is greater than or equal to zero, then the gateway accepts the routing update, sends an acknowledgment (see Appendix A) to the neighbor containing the sequence number S, and replaces the Receive Sequence Number, R, with S. If this value (S-R) is less than zero, the gateway rejects the routing update and sends a negative acknowledgment [Appendix A] to the neighbor with sequence number R.

The gateway has a Send Sequence Number, N, for sending routing updates to all of its neighbors. This sequence number

can be initialized to any value. The Send Sequence Number is incremented each time a new routing update is created. On receiving an acknowledgment for a routing update, the gateway subtracts the sequence number acknowledged, A, from the Send Sequence Number, N. If the value (N-A) is non-zero, then an old routing update is being acknowledged. The gateway continues to retransmit the most recent routing update to the neighbor that sent the acknowledgment. If (N-A) is zero, the routing update has been acknowledged. Note that only the most recent routing update must be acknowledged; if a second routing update is generated before the first routing update is acknowledged, only the second routing update must be acknowledged.

If a negative acknowledgment is received, the gateway subtracts the sequence number negatively acknowledged, A, from its Send Sequence Number, N. If this value (N-A) is less than zero, then the gateway replaces its Send Sequence Number, N, with the sequence number negatively acknowledged plus one, A+1, and retransmits the routing update to all of its neighbors. If (N-A) is greater than or equal to zero, then the gateway continues to retransmit the routing update using sequence number N. In order to maintain the correct sequence numbers at all gateways, routing updates must be retransmitted to all neighbors if the Send

Sequence Number changes, even if the routing information does not change.

The gateway retransmits routing updates periodically until they are acknowledged and whenever its Send Sequence Number changes. The gateway sends routing updates only to neighbors that are in the "up" state.

4.4.4 Computing Routes

A routing update contains a list of networks that are reachable through this gateway, and the distance in "number of hops" to each network mentioned. The routing update only contains information about a network if the gateway believes that it is as close or closer to that network then the neighbor which is to receive the routing update. The network address may be an internet class A, B, or C address.

The information inside a routing update is processed as follows. The gateway contains an N x K distance matrix, where N is the number of networks and K is the number of neighbor gateways. An entry in this matrix, represented as dm(I,J), is the distance to network I from neighbor J as reported in the most

recent routing update from neighbor J. The gateway also contains a vector indicating the connectivity between itself and its neighbor gateways. The values in this vector are computed as discussed above (see Section 4.4.2, Determining Connectivity to Neighbors). The value of the Jth entry of this vector, which is the connectivity between the gateway and the Jth neighbor, is represented as d(J).

The gateway copies the routing update received from the Jth neighbor into the appropriate row of the distance matrix, then updates its routes as follows. The gateway calculates a minimum distance vector which contains the minimum distance to each network from the gateway. The Ith entry of this vector, represented as MinD(I) is:

MinD(I) = minimum over all neighbors of d(J) + dm(I,J)

where d(J) is the distance between the gateway and the Jth neighbor, and dm(I,J) is the distance from the Jth neighbor to the Ith network. If the Ith network is attached to the gateway and the gateway can send and receive traffic on its network interface (see Section 4.4.2), then the gateway sets the Ith entry of the minimum distance vector to zero.

Using the minimum distance vector, the gateway computes a list of neighbor gateways through which to send traffic to each network. The entry for a given network contains one of the neighbors that is the minimum distance away from that network.

After updating its routes to the networks, the gateway computes the new routing updates to be sent to its neighbors. The gateway reports a network to a neighbor only if it is as close to or closer to that network than its neighbor. For each network I, the routing update contains the address of the network and the minimum distance to that network which is MinD(I).

Finally, the gateway must determine whether it should send routing updates to its neighbors. The gateway sends new updates to its neighbors if every one of the following three conditions occurs: 1) one of the gateway's interfaces changes state, 2) one of the gateway's neighbor gateways changes state, and 3) the gateway receives a routing update from a neighbor that is different from the update that it had previously received from that neighbor. The gateway sends routing updates only to neighbors that are currently in the "up" state.

The gateway requests a routing update from neighbors that are in the "up" state, but from which it has not yet received a

routing update. Routing updates are requested by setting the appropriate bit in the routing update being sent [Appendix A]. Similarly, if a gateway receives from a neighbor a routing update in which the bit requesting a routing update is set, the gateway sends the neighbor the most recent routing update.

4.4.5 Non-Routing Gateways

A Non-routing Gateway is a gateway that forwards internet traffic, but does not implement the GGP routing algorithm. Networks that are behind a Non-routing Gateway are known a-priori to Routing Gateways. There can be one or more of these networks which are considered to be directly connected to the Non-routing Gateway. A Routing Gateway will forward a datagram to a Non-routing Gateway if it is addressed to a network behind the Non-routing Gateway. Routing Gateways currently do not send Redirects for Non-routing Gateways. A Routing Gateway will always use another Routing Gateway as a path instead of a Non-routing Gateways if both exist and are the same number of hops away from the destination network. The Non-routing Gateways path will be used only when the Routing Gateway path is down; when the Routing Gateway path comes back up, it will be used again.

4.4.6 Adding New Neighbors and Networks

Gateways dynamically add routing information about new neighbors and new networks to their tables. The gateway maintains a list of neighbor gateway addresses. When a routing update is received, the gateway searches this list of addresses for the Internet source address of the routing update message. If the Internet source address of the routing update is not contained in the list of neighbor addresses, the gateway adds this address to the list of neighbor addresses and sets the neighbor's connectivity status to "down." Routing updates are not accepted from neighbors until the GGP polling mechanism has determined that the neighbor is up.

This strategy of adding new neighbors requires that one gateway in each pair of neighbor gateways must have the neighbor's address configured in its tables. The newest gateway can be given a complete list of neighbors, thus avoiding the need to re-configure older gateways when new gateways are installed.

Gateways obtain routing information about new networks in several steps. The gateway has a list of all the networks for which it currently maintains routing information. When a routing update is received, if the routing update contains information

about a new network, the gateway adds this network to the list of networks for which it maintains routing information. Next, the gateway adds the new network to its distance matrix. The distance matrix comprises the is the matrix of distances (number of hops) to networks as reported in routing updates from the neighbor gateways. The gateway sets the distance to all new networks to "infinity," and then computes new routes and new routing updates as outlined above.

4.5 Exterior Gateway Protocol

The Exterior Gateway Protocol (EGP) is used to permit other gateways and gateway systems to pass routing information to the DARPA Internet gateways. The use of the EGP permits the user to perceive all of the networks and gateways as part of one total Internet system, even though the "exterior" gateways are disjoint and may use a routing algorithm that is different and not compatible with that used in the "interior" gateways. The important elements of the EGP are:

o Neighbor Acquisition

> The procedure by which a gateway requests that it become a neighbor of another gateway. This is used when a gateway wants to become a neighbor of another in order to pass

routing information. This includes the capability to accept or refuse the request.

o Neighbor Up/Down

The procedure by which a gateway decides if another gateway is up or down.

o Network Reachability Information

The facility used to pass routing and neighbor information between gateways.

o Gateway Going Down

The ability of a gateway to inform other gateways that it is going down and no longer has any routes to any other networks. This permits a gateway to go down in an orderly way without disrupting the rest of the Internet system.

A complete description of the EGP can be found in ~~IEN-209,~~ RFC-827 the "Exterior Gateway Protocol" [10].

5 GATEWAY SOFTWARE

The DARPA Internet Gateway runs under the MOS operating system [9] which provides facilities for:

- o Multiple processes
- o Interprocess communication
- o Buffer management
- o Asynchronous input/output
- o Shareable real-time clock

There is a MOS process for each network that the gateway is directly connected to. A data structure called a NETBLOCK contains variables of interest for each network and pointers to local network routines. Network processes run common gateway code while network-specific functions are dispatched to the routines pointed to in the NETBLOCK. There are also processes for gateway functions which require their own timing, such as GGP and HMP.

5.1 Software Structure

The gateway software can be divided conceptually into three parts: MOS Device Drivers, Network software, and Shared Gateway software.

5.1.1 Device Drivers

The gateway has a set of routines to handle sending and receiving data for each type of hardware interface. There are routines for initialization, initiation, and interruption for both the transmit and receive sides of a device. The gateway supports the following types of devices:

a) ACC LSI-11 1822
b) DEC IMP11a 1822
c) ACC LHDH 1822
d) ACC VDH11E
e) ACC VDH11C
f) Proteon Ring Network
g) RSRE HDLC
h) Interlan Ethernet
i) BBN Fibernet
j) ACC XQ/CP X.25 **
k) ACC XQ/CP HDH **

5.1.2 Network Software

For each connected network, the gateway has a set of eight routines which handle local network functions. The network routines and their functions are described briefly below.

←←←←←←←←←←←←←←←
** Planned, not yet supported.

Up.net	Perform local network initialization such as flapping the 1822 ready line.
Sg.net	Handle specific local network timing functions such as timing out 1822 Destination Deads.
Rc.net	A message has been received from the network interface. Check for any input errors.
Wc.net	A message has been transmitted to the network interface. Check for any output errors.
Rs.net	Set up a buffer (or buffers) to receive messages on the network interface.
Ws.net	Transmit a message to the network interface.
Hc.net	Check the local network header of the received message. Perform any local network protocol tasks.
Hb.net	Rebuild the local network header.

There are network routines for the following types of networks:

- Arpanet (a,b,c,k)
- Satnet (d,e,k)
- Proteon Ring Network (f)
- Packet Radio Network (a,b,c)
- Rsre HDLC Null Network (g)
- Ethernet (h)
- Fibernet (i)
- Telenet X.25 (j) **

Note: The letters in parentheses refer to the device drivers used

←←←←←←←←←←←←←←

** Planned, not yet supported.

for each type of network as described in the previous section.

5.1.3 Shared Gateway Software

The internet processing of a datagram is performed by a body of code which is shared by the network processes. This code includes routines to check the IP header, perform IP fragmentation, calculate the IP checksum, forward a datagram, and implement the routing, monitoring, and error reporting protocols.

5.2 Gateway Processes

5.2.1 Network Processes

When the gateway starts up, each network process calls its local network initialization routine and read start routine. The read start routine sets up two maximum network size buffers for receiving datagrams. The network process then waits for an input complete signal from the network device driver.

When a message has been received, the MOS Operating System signals the appropriate network process with an input complete signal. The network process wakes up and executes the net read

complete routine. After the message has been processed, the network process waits for more input.

The net read complete routine is the major message processing loop in the gateway. The following actions are performed when a message has been received:

- Call Local Network Read Complete Routine
- Start more reads
- Check local Network Header
- Check Internet header
- Check if datagram is for the gateway
- Forward the datagram if necessary
- Send ICMP error message if necessary.

5.2.2 GGP Process

The GGP process periodically sends GGP echos to each of the gateway's neighbors to determine neighbor connectivity, and sends interface status messages addressed to itself to determine network connectivity. The GGP process also sends out routing updates when necessary. The details of the algorithms currently implemented by the GGP process are given in Section 4.4, Gateway-to-Gateway Protocol, and in Appendix C.

5.2.3 HMP Process

The HMP process handles timer-based gateway statistics collection and the periodic transmission of traps.

APPENDIX A. GGP Message Formats

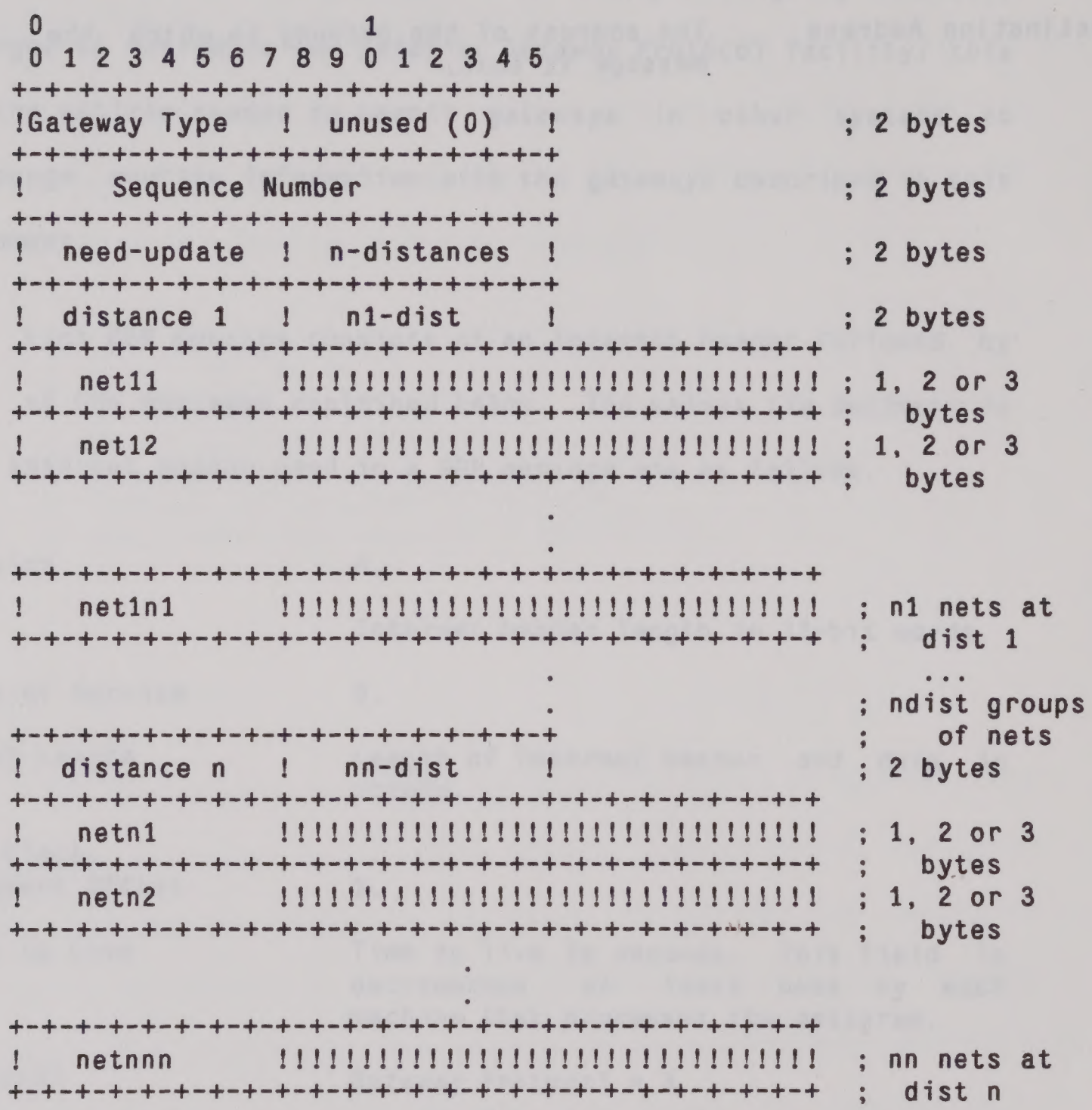
Note that the GGP protocol is currently undergoing extensive changes to introduce the Exterior Gateway Protocol facility; this is the vehicle needed to permit gateways in other systems to exchange routing information with the gateways described in this document.

Each GGP message consists of an Internet header followed by one of the messages explained below. The values (in decimal) in the Internet header used in a GGP message are as follows.

Version	4.
IHL	Internet header length in 32-bit words.
Type of Service	0.
Total Length	Length of Internet header and data in octets.
ID, Flags, Fragment Offset	0.
Time to Live	Time to live in seconds. This field is decremented at least once by each machine that processes the datagram.
Protocol	Gateway Protocol = 3.
Header Checksum	The 16 bit one's complement of the one's complement sum of all 16-bit words in the header. For computing the checksum, the checksum field should be zero.

Source Address	The address of the gateway's interface from which the message is sent.
Destination Address	The address of the gateway to which the message is sent.

ROUTING UPDATE



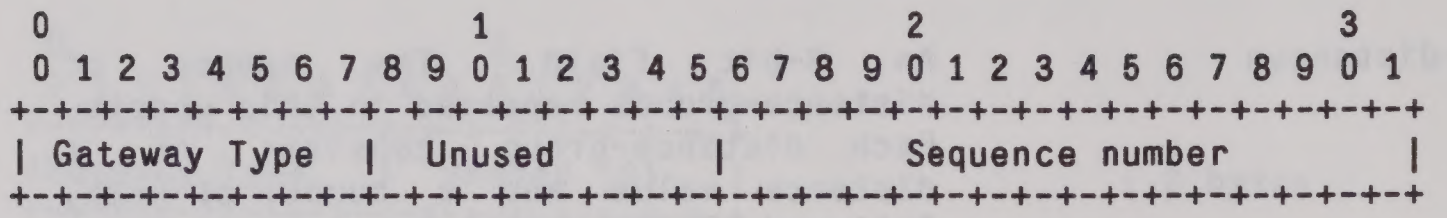
Gateway Type — 12 (decimal)

Sequence Number — The 16-bit sequence number used to identify routing updates.

need-update — An 8-bit field. This byte is set to 1

	if the source gateway requests a routing update from the destination gateway, and set to 0 if not.
n-distances	An 8-bit field. The number of distance-groups reported in this update. Each distance-group consists of a distance value and a number of nets, followed by the actual net numbers which are reachable at that distance. Not all distances need be reported.
distance 1	hop count (or other distance measure) which applies to this distance-group.
n1-dist	number of nets which are reported in this distance-group.
net11	1, 2, or 3 bytes for the first net at distance "distance 1".
net12	second net
...	
net1n1	etc.

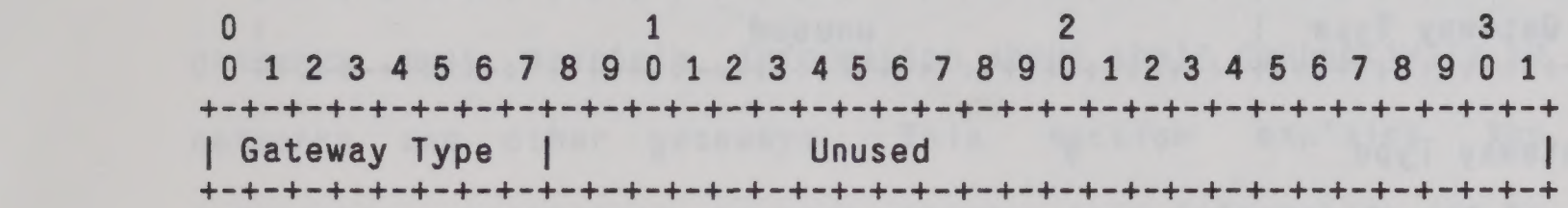
ACKNOWLEDGMENT or NEGATIVE ACKNOWLEDGMENT



Gateway Type — Acknowledgments are type 2. Negative acknowledgments are type 10.

Sequence Number — The 16-bit sequence number that the gateway is acknowledging or negatively acknowledging.

GGP ECHO and ECHO REPLY



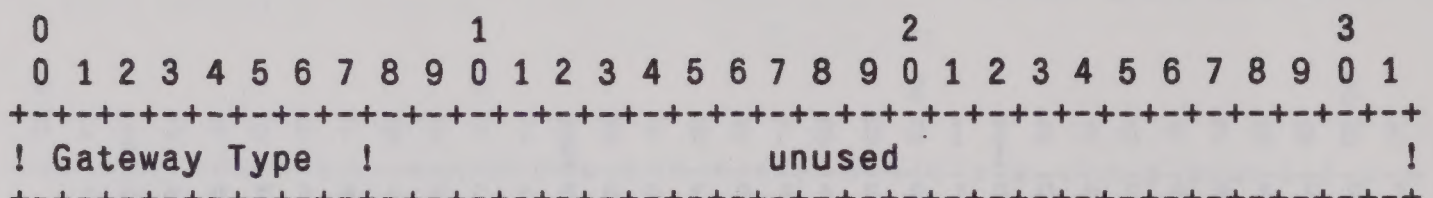
Gateway Type

8 for echo message; 0 for echo reply.

Source Address

In an echo message, this is the address of the gateway on the same network as the neighbor to which it is sending the echo message. In an echo reply message, the source and destination addresses are simply reversed, and the remainder is returned unchanged.

NETWORK INTERFACE STATUS



Gateway Type 9

Source Address
Destination Address — The address of the gateway's network interface. The gateway can send Net Interface Status messages to itself to determine if it is able to send and receive traffic on its network interface.

APPENDIX B. Information Maintained by Gateways

In order to implement the shortest-path routing algorithm, gateways must maintain information about their connectivity to networks and other gateways. This section explains the information maintained by each gateway; this information can be organized into the following tables and variables.

- Number of Networks

  The number of networks for which the gateway maintains routing information and to which it can forward traffic.

- Number of Neighbors

  The number of neighbor gateways with which the gateway exchanges routing information.

- Gateway Addresses

  The addresses of the gateway's network interfaces.

- Neighbor Gateway Addresses

  The address of each neighbor gateway's network interface that is on the same network as this gateway.

- Neighbor Connectivity Vector

  A vector of the connectivity between this gateway and each of its neighbors.

- Distance Matrix

  A matrix of the routing updates received from the neighbor gateways.

- Minimum Distance Vector

  A vector containing the minimum distance to each network.

- Routing Updates from Non-Routing Gateways

  The routing updates that would have been received from each non-routing neighbor gateway which does not participate in this routing strategy.

- Routing Table

  A table containing, for each network, a list of the neighbor gateways on a minimum-distance route to the network.

- Send Sequence Number

  The sequence number that will be used to send the next routing update.

- Receive Sequence Numbers

  The sequence numbers that the gateway received in the last routing update from each of its neighbors.

- Received Acknowledgment Vector

  A vector indicating whether or not each neighbor has acknowledged the sequence number in the most recent routing update sent.

APPENDIX C. GGP Events and Responses

The following list shows the GGP events that occur at a gateway and the gateway's responses. The variables and tables referred to are listed above.

o Connectivity to an attached network changes.

  a. Update the Minimum Distance Vector.
  b. Recompute the Routing Updates.
  c. Recompute the Routing Table.
  d. If any routing update has changed, send the new routing updates to the neighbors.

o Connectivity to a neighbor gateway changes.

  a. Update the Neighbor Connectivity Vector.
  b. Recompute the Minimum Distance Vector.
  c. Recompute the Routing Updates.
  d. Recompute the Routing Table.
  e. If any routing update has changed, send the new routing updates to the neighbors.

o A Routing Update message is received.

  a. Compare the Internet source address of the Routing Update message to the Neighbor Addresses. If the address is not on the list, add it to the list of Neighbor Addresses, increment the Number of Neighbors, and set the Receive Sequence Number for this neighbor to the sequence number in the Routing Update message.

  b. Compare the Receive Sequence Number for this neighbor to the sequence number in the Routing Update message to determine whether or not to accept this message. If the message is rejected, send a Negative Acknowledgment message. If the message is accepted, send an Acknowledgment message and proceed with the following steps.

c. Compare the networks reported in the Routing Update message to the Number of Networks. If new networks are reported, enter them in the network vectors, increase the number of networks, and expand the Distance Matrix to account for the new networks.

d. Copy the routing update received into the appropriate row of the Distance Matrix.

e. Recompute the Minimum Distance Vector.

f. Recompute the Routing Updates.

g. Recompute the Routing Table.

h. If any routing update has changed, send the new routing updates to the neighbors.

o An Acknowledgment message is received.

Compare the sequence number in the message to the Send Sequence Number. If the Send Sequence Number is acknowledged, update the entry in the Received Acknowledgment Vector for the neighbor that sent the acknowledgment.

o A Negative Acknowledgment message is received.

Compare the sequence number in the message to the Send Sequence Number. If necessary, replace the Send Sequence Number, and retransmit the routing updates.

REFERENCES

[1] Postel, J. (ed.), "Internet Protocol - DARPA Internet Program Protocol Specification," RFC 791, USC/Information Sciences Institute, September 1981.

[2] Strazisar, V., "Gateway Routing: An Implementation Specification," IEN-30, Bolt Beranek and Newman Inc., August 1979.

[3] Strazisar, V., "How to Build a Gateway," IEN-109, Bolt Beranek and Newman Inc., August 1979.

[4] Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification," RFC 792, USC/Information Sciences Institute, September 1981.

[5] Postel, J., "Assigned Numbers," RFC 790, USC/Information Sciences Institute, September 1981.

[6] Littauer, B., Huang, A., Hinden, R., "A Host Monitoring Protocol," IEN-197, Bolt Beranek and Newman Inc., September 1981.

[7] Santos, P., Chalstrom, H., Linn, J., Herman, J., "Architecture of a Network Monitoring, Control and Management System," Proc. of the 5th Int. Conference on Computer Communication, October 1980.

[8] Haverty, J., "XNET Formats for Internet Protocol Version 4," IEN-158, Bolt Beranek and Newman Inc., October 1980.

[9] Mathis, J., Klemba, K., Poggio, "TIU Notebook- Volume 2, Software Documentation," SRI, May 1979.

[10] Rosen, E., "Exterior Gateway Protocol," ~~IEN-209~~ RFC-827, Bolt Beranek and Newman Inc., August 1982.